Appln No. 09/916,557
Amdt date September 23, 2005
Reply to Office action of July 26, 2005

REMARKS/ARGUMENTS

Claims 1-22 are currently pending in this application. Claims 1 and 11 have been amended. The amendments find full support in the original specification, claims, and drawings. No new matter has been added. In view of the above amendments and remarks that follow, reconsideration, reexamination, and an early indication of allowance of claims 1-22 are respectfully requested.

Claims 1-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over European Publication No. EP 0895164 (Vano) in view of in view of Schneier ("Applied Cryptography," Ch. 17, pp. 397-398). Applicant respectfully traverses this rejection.

Claim 1 has been amended to recite that "the state memory is initialized via hardware with an incrementing pattern without loading the incrementing pattern from an external memory." (Emphasis added). Applicant respectfully submits that a person of skill in the art would not have been motivated to modify Vano's state register (reference number 554), which the Examiner contends reads on the claimed "state memory," to cause the register to be "initialized via hardware with an incrementing pattern without loading the incrementing pattern from an external memory." In Vano's system, the state register 554 is included in an execution unit (EU). (Col. 5, lines 20-22; lines 46-53). Vano teaches that the EU, and thus, the state register included in the EU, is "preferably pre-configured during a channel program setup and configuration process." (Col. 5, lines 17-19). During this channel program setup and configuration process, an external controller 10 performs a setup and configuration procedure for a channel program which prepares the configurable cryptographic engine (CCE) for processing data units. The channel program "is preferably created by means external to the CCE." (Col. 4, lines 19-20). During the setup and configuration procedure, the external controller then

loads the externally created channel program from an external memory 12. (Col. 4, lines 27-32). This implies that data for configuring the state register is also created externally and loaded from the external memory, which is contrary to the claimed limitation that "the state memory is initialized via hardware with an incrementing pattern without loading the incrementing pattern from an external memory." (Emphasis added). The Examiner has not indicated why a person of skill in the art would have been motivated to go against the express teaching of Vano.

The Examiner contends that the fact that the initialization is done with or without loading the incrementing pattern from an external memory is "design choice," and that one of ordinary skill in the art could "easily implement the RC4 algorithm by initializing the incrementing pattern either with or without loading the incrementing pattern from an external memory."

By the above arguments, the Examiner appears to contend that it is common knowledge to initialize the incrementing pattern "without loading the incrementing pattern from an external memory." If it is the Examiner's position that it is also common knowledge to initialize the state memory "via hardware . . . without loading the incrementing pattern from an external memory," Applicant respectfully requests that the Examiner provide some form of evidence to support the assertion of common knowledge. (See, M.P.E.P. § 2144.03.) The M.P.E.P. makes it clear that "[i]t is never appropriate to rely solely on 'common knowledge' in the art without evidentiary support in the record, as the principal evidence upon which a rejection was based." Applicant respectfully submits that none of the cited references provide any sort of evidence to support that it is common knowledge to initialize the incrementing pattern "via hardware . . . without loading the incrementing pattern from an external memory." In fact, Vano provides contrary evidence via its disclosure that such initialization data

would be loaded from an external memory during a setup and configuration procedure.

Schneier also fails to teach or suggest that "the state memory is initialized via hardware with an incrementing pattern without loading the incrementing pattern from an external memory." The Examiner relies on Schneier's disclosure of filling an S-box linearly to contend that it teaches a state memory that is initialized with an incrementing pattern. However, as discussed by Schneier on page 349 (a copy of which is enclosed for the Examiner's reference), an S-box is simply a lookup table made up of arrays. There is no teaching or suggestion in any of the cited references that such arrays of a lookup table would be initialized "via hardware with an incrementing pattern without loading the incrementing pattern from an external memory." (Emphasis added). Accordingly, claim 1 is now in condition for allowance.

Claim 11 includes limitations that are similar to the limitations of claim 1 which make claim 1 allowable. Thus, claim 11 is allowable at least for the reasons indicated for claim 1. In addition, claim 11 recites "performing a shuffling operation on the fly while concurrently retrieving a secret key associated with the data, wherein the shuffling operation includes moving each of the plurality of state memory values based upon the secret key," which is not taught nor suggested by Vano. The Examiner contends, however, that Vano's permuters select bits from state register, that the state register contains channel program, and that the key is also contained in the channel program. Vano's state registers, however, simply contain "channel program state information." (Col. 6, lines 19-22). The actual channel programs are not stored in the state registers, but rather, in microcode memory 200 or 202. (Col. 4, lines 29-34). Also, any key information in Vano is stored in active and shadow "variable registers 556 & 557." (Col. 6, lines 23-30). The permuter does not

select bits from these variable registers. Thus Vano's permuter does not execute the step of "performing a shuffling operation on the fly while concurrently retrieving a secret key associated with the data" as is recited in claim 11. Thus, claim 11 is also in condition for allowance due to this additional limitation.

Claims 2-10 and 12-22 are also in condition for allowance because they depend on an allowable base claim, and for the additional limitations that they contain.

In view of the above amendments and remarks, reexamination, reconsideration and an early indication of allowance of claims 1-22 are respectfully requested.

Respectfully submitted,
CHRISTIE, PARKER & HALE, LLP

By _____
Josephine E. Chang
Reg. No. 46,083
626/795-9900

JEC/lal
Enclosure: Schneier, *Applied Cryptography*, Second Edition, pg. 349

LAL PAS636975.1-*-09/23/05 11:23 AM

"... the best introduction to cryptography I've ever seen. The book the National Security Agency wanted never to be published..." —Wired Magazine

MORE THAN 100,000 COPIES SOLD
SECOND EDITION

APPLIED CRYPTOGRAPHY

Protocols, Algorithms, and Source Code in C

BRUCE SCHNEIER

Publisher: Katherine Schowalter
Editor: Phil Sutherland
Assistant Editor: Allison Roarty
Managing Editor: Robert Aronds
Text Design & Composition: North Market Street Graphics

Library of Congress Cataloging-in-Publication Data:

Schneier, Bruce
Applied Cryptography Second Edition : protocols, algorithms, and source code in C
/ Bruce Schneier.
p. cm.
Includes bibliographical references (p. 675).
ISBN 0-471-12845-7 (cloth : acid-free paper). — ISBN
0-471-11709-9 (paper : acid-free paper)
1. Computer security. 2. Telecommunication—Security measures.
3. Cryptography. I. Title.
QA76.9.A25S35 1996
005.8'2—dc20 95-12398
CIP

Printed in the United States of America
20 19 18 17 16 15 14 13 12

tials seem to apply only to ciphers with a small number of rounds, but partial differentials combine nicely with differentials.

Linear cryptanalysis is newer, and is still being improved. Notions of key ranking [1019] and multiple approximations [811,812] have been defined. Other work that extends the idea of linear cryptanalysis can be found in [1270]; [938] tries to combine linear and differential cryptanalysis into one attack. It is unclear what design techniques will protect against these sorts of attacks.

Knudsen has made some progress, considering some necessary (but not perhaps sufficient) criteria for what he calls **practically secure Feistel networks**: ciphers that resist both linear and differential cryptanalysis [857]. Nyberg introduced in linear cryptanalysis an analogy to the concept of differentials from differential cryptanalysis [1180].

Interestingly enough, there seems to be a duality between differential and linear cryptanalysis. This duality becomes apparent both in the design of techniques to construct good differential characteristics and linear approximations [164,1018], and also in the design criteria for making algorithms that are secure against both attacks [307]. Exactly where this line of research will lead is still unknown. As a start, Daemen has developed an algorithm-design strategy based on linear and differential cryptanalysis [402].

S-Box Design

The strength of various Feistel networks—and specifically their resistance to differential and linear cryptanalysis—is tied directly to their S-boxes. This has prompted a spate of research on what constitutes a good S-box.

An S-box is simply a substitution: a mapping of m-bit inputs to n-bit outputs. Previously I talked about one large lookup table of 64-bit inputs to 64-bit outputs; that would be a 64*64-bit S-box. An S-box with an m-bit input and an n-bit output is called a **m*n-bit S-box**. S-boxes are generally the only nonlinear step in an algorithm; they are what give a block cipher its security. In general, the bigger they are, the better.

DES has eight different 6*4-bit S-boxes. Khufu and Khafre have a single 8*32-bit S-box, LOKI has a 12*8-bit S-box, and both Blowfish and CAST have 8*32-bit S-boxes. In IDEA the modular multiplication step is effectively the S-box; it is a 16*16-bit S-box. The larger this S-box, the harder it is to find useful statistics to attack using either differential or linear cryptanalysis [653,729,1626]. Also, while random S-boxes are usually not optimal to protect against differential and linear attacks, it is easier to find strong S-boxes if the S-boxes are larger. Most random S-boxes are nonlinear, nondegenerate, and have strong resistance to linear cryptanalysis—and the fraction that does not goes down rapidly as the number of input bits decreases [1185,1186,1187].

The size of m is more important than the size of n. Increasing the size of n reduces the effectiveness of differential cryptanalysis, but greatly increases the effectiveness of linear cryptanalysis. In fact, if $n \geq 2^m - m$, then there is definitely a linear relation of the input and output bits of the S-box. And if $n \geq 2^m$, then there is a linear relation of only the output bits [164].